DIFF B/W CHECKED AND UNCHECKED EXCEPTION :

09 February 2025 11:57

Misconception	Reality
"Compile-time exceptions occur at compile-time."	Compile-time exceptions are **detected** at compile time, but they **occur** only at runtime.
"Runtime exceptions occur at runtime."	This is correct! Runtime exceptions are both **detected** and **occur** at runtime.

➢ **Important Takeaway**

- Every **red line error** = **Compile-Time Error**.
- But not every red line error is a **Compile-Time Exception**.
- **Compile-Time Exceptions** specifically refer to **checked exceptions**, which you must explicitly handle using try-catch or throws.

Checked Exception / Compile Time Exception	Unchecked Exception / Runtime Exception
1. Checked Exceptions are the exceptions that are checked and handled at compile time.	1. Unchecked Exceptions are the exceptions that are not checked at compiled time.
2. The program gives a compilation error if a method throws a checked exception.	2. The program compiles fine because the compiler is not able to check the exception.
3. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword.	3. A method is not forced by compiler to declare the unchecked exceptions thrown by its implementation. Generally, such methods almost always do not declare them, as well.
4. A checked exceptions occur when the chances of failure are too high.	4. Unchecked exception occurs mostly due to programming mistakes.
5. They are direct subclass of Exception class but do not inherit from RuntimeException.	5. They are direct subclass of RuntimeException class.